entropy

MDPI

Article

A First Step to the Categorical Logic of Quantum Programs

Xin Sun [1,*] and Feifei He [2]

[1] Department of foundation of computer science, the John Paul II Catholic University of Lublin, 20-950 Lublin, Poland
[2] Institute of logic and cognition, Sun Yat-sen University, Guangzhou 510970, China; hliheng@gmail.com
* Correspondence: xin.sun.logic@gmail.com

Received: 2 December 2019; Accepted: 7 January 2020; Published: 24 January 2020

Abstract: The long-term goal of our research is to develop a powerful quantum logic which is useful in the formal verification of quantum programs and protocols. In this paper we introduce the basic idea of our categorical logic of quantum programs (CLQP): It combines the logic of quantum programming (LQP) and categorical quantum mechanics (CQM) such that the advantages of both LQP and CQM are preserved while their disadvantages are overcome. We present the syntax, semantics and proof system of CLQP. As a proof-of-concept, we apply CLQP to verify the correctness of Deutsch's algorithm and the concealing property of quantum bit commitment.

Keywords: quantum logic; quantum computing; category theory

1. Introduction

Quantum programs and quantum protocols are two pillars of quantum computing. The exponential speedup provided by Shor's factoring algorithm and the quadratic speedup provided by Grover's search algorithm over their classical counterparts has brought quantum computing into the limelight. The unconditional security offered by quantum protocols such as quantum key distribution has grabbed strong interests from both the academic and the industrial community. Designing quantum programs and protocols is an error-prone task due to the counter-intuitive nature of quantum systems. Therefore, verification techniques for quantum programs and quantum protocols will be indispensable in the coming era of quantum computation. For example, a number of quantum process calculi [1–4] have been proposed for the formal verification of quantum protocols. Some quantum logics have been developed to verify both quantum programs and quantum protocols.

Quantum logic began with Birkhoff and von Neumann [5]. Traditional quantum logic [5–10] focuses on the order-theoretic structure of testable properties in the quantum world and is based on the lattice of closed subspaces of a (usually infinite dimensional) Hilbert space. The success of quantum computation and quantum information has inspired new quantum logics [11,12] which are based on *finite* dimensional Hilbert spaces, such as qubits. Brunet and Jorrand [13] proposed to extend the Birkhoff–von Neumann quantum logic to reasoning about quantum programs. Chadha et al. [14] presented a first attempt to develop a Hoare-like logic for quantum programs. Some useful proof rules for reasoning about quantum programs were introduced by Fenget et al. [15]. A full-fledged Hoare-like logic for both partial and total correctness of quantum programs was established in Ying [16].

The logic for quantum programs (LQP) [17–21] is an extension of traditional quantum logic and quantum Hoare logic. It has been used to verify quantum search algorithms [20], quantum leader election [20], quantum key distribution [21] and quantum voting [22]. The expressive power of LQP is largely determined by the constant symbols it incorporates. There is no systematic study of constant symbols in the literature of LQP. In Baltag and Smets [18], the authors chose the following unitary

operators as constant symbols: X, Z, H and CNOT. In Rad et al. [22], Bell states are used as constant symbols. Those operators are not universal. Therefore, there are still many quantum states and operators that cannot be expressed in LQP. Another limitation of the presentation of constant symbols in LQP is the missing satisfying axiomatization. Different axioms for different constant symbols are introduced, depending on which programs/protocols are to be verified. These two limitations make LQP not a convenient tool in the formal verification of quantum programs and protocols: To verify a quantum program or protocol in LQP, we have to first find an appropriate set of constant symbols to express the program/protocol, then we still need to introduce a set of axioms for these constant symbols such that some desired properties of the targeting program/protocol can be proved axiomatically. This procedure usually consumes a lot of time and intelligence. We believe that LQP cannot be successful in formal verification if these two limitations are not overcome.

In this paper, we will overcome these limitations by extending LQP to the categorical logic of quantum programs (CLQP). CLQP is a combination of LQP and categorical quantum mechanics (CQM) [23–26]. The main feature of the construction of CLQP is the representation of constant symbols of LQP by morphisms in the ZX-calculus, a graphical calculus of CQM. Inherited from the universality of the ZX-calculus [24], CLQP has stronger expressive power than LQP. CLQP also inherits the graphical axiomatization of the ZX-calculus such that many properties of a quantum program and protocols can be proved concisely and graphically, when the program/protocol is specified in CLQP. On the other hand, CLQP preserves the various logical operations (for example, boolean connectives, programs constructors, epistemic modality and probabilistic modality) of LQP. These operations allow us to express various properties that are not expressible in CQM. In a nutshell, CLQP keeps the advantages of both LQP and CQM, while overcoming their limitations. These features make CLQP a powerful tool for the formal specification and verification of quantum programs and protocols.

The structure of this paper is as follows. In Section 2 we will introduce the syntax, semantics and axiomatization of CLQP. Then in Section 3 we apply CLQP to the verification of quantum programs and protocols. We conclude this paper with future work in Section 4.

2. Categorical Logic for Quantum Programs

2.1. Syntax

For each natural number $n \geq 1$, we build the n-qubit categorical logic of quantum programs CLQP_n. To build the language of CLQP_n, we are given the following: A finite set of natural numbers $N = \{1, \ldots, n\}$, a countable set of propositional variables $\mathbb{P}$, a countable set of operational variables $\mathbb{O}$, constants symbols of propositions and unitary operations built from the ZX-calculus.

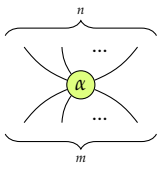
2.1.1. Constants from Categorical Quantum Mechanics

Categorical quantum mechanics is the study of quantum computation and quantum foundations using category theory. The ZX-calculus, a graphical language of quantum computation developed in the framework of CQM, is introduced by Coecke and Duncan [24,27]. It is founded on a dagger symmetric monoidal category (†-SMC) $\mathfrak{C}$. (A concise introduction to †-SMC is provided in the Appendix A.)

The objects of $\mathfrak{C}$ are natural numbers: $0, 1, 2, \cdots$; the tensor of objects is just the addition of numbers: $m \otimes n = m + n$. 0 is the unit object of $\mathfrak{C}$. In the matrix interpretation of the ZX-calculus, an object n is interpreted as the 2^n dimensional Hilbert space $\mathbb{C}^{2^n}$. An identity morphism of $\mathfrak{C}$ is interpreted as the identity map on the corresponding Hilbert space. A swap morphism $\sigma_{m,n}$ is interpreted as the map $SWAP_{m,n}$ to which we have $SWAP_{m,n}(|a\rangle|b\rangle) = |b\rangle|a\rangle$ for all $|a\rangle \in \mathbb{C}^{2^m}$ and $|b\rangle \in \mathbb{C}^{2^n}$. Apart from the identity morphisms and swap morphisms, the following are also the basic morphisms of $\mathfrak{C}$:

1. Z-spiders $Z_m^n(\alpha) : m \to n$, for every real number $\alpha \in \mathbb{R}$, of which the matrix interpretation and graphical representation are respectively

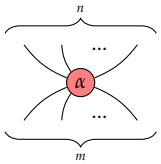
$$|0\rangle^{\otimes n}\langle 0|^{\otimes m} + e^{i\alpha}|1\rangle^{\otimes n}\langle 1|^{\otimes m} \text{ and } \quad .$$



We call α the phase of $Z^n_m(\alpha)$. The graph is read from bottom to top. We consider the wires at the bottom as input and those on the top as output. We usually omit the phase in the graphical representation when it is 0.

2. X-spiders $X^n_m(\alpha) : m \to n$, for every real number $\alpha \in \mathbb{R}$, of which the matrix interpretation and graphical representation are respectively

$$|+\rangle^{\otimes n}\langle +|^{\otimes m} + e^{i\alpha}|-\rangle^{\otimes n}\langle -|^{\otimes m} \text{ and } \quad .$$



Similarly, we call α the phase of $X^n_m(\alpha)$.

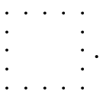
3. The identity morphism $Id : 1 \to 1$. The matrix interpretation of Id is the identity map on the Hilbert space $\mathbb{C}^2$. The graphical representation of Id is a straight line

.

4. The H-box $H : 1 \to 1$, of which the matrix interpretation and graphical representation are respectively

$$\frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \text{ and } \quad .$$

5. $e : 0 \to 0$ is interpreted as the number 1. The graphical representation of e is the empty graph



.

6. SWAP morphism $\sigma : 2 \to 2$, of which the matrix interpretation and graphical representation are respectively

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \text{ and } \quad .$$

7. Bell state $\beta : 0 \to 2$, of which the matrix interpretation and graphical representation are respectively

$$\begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix} \text{ and } \quad .$$

8. Bell effect $\beta : 2 \to 0$, of which the matrix interpretation and graphical representation are respectively

$\begin{bmatrix} 1 & 0 & 0 & 1 \end{bmatrix}$ and  .

The morphisms of $\mathfrak{C}$ are generated by applying sequential and parallel composition of the basic morphisms, or by applying the dagger operation † to a morphism. The matrix interpretation of sequential and parallel composition of morphisms is the matrix production and tensor product, respectively. The graphical representation of sequential composition of morphisms is to put one graph on top of another, while the parallel composition of morphisms is represented by putting graphs side-by-side. The matrix interpretation of † is the adjoint operation. The graphical representation of † is to turn the graph upside-down, meanwhile changing the sign of phases that appear in the graph.

The ZX-calculus is a universal language for quantum computing in the sense that it can represent all linear maps between qubit systems. Another impressive feature of the ZX-calculus is that it admits a sound and complete axiomatization [28–30] to derive equations between morphisms. The axiomatization of the ZX-calculus will play an important role in the axiomatization of CLQP. There are two kinds of axioms to determine whether two morphisms of $\mathfrak{C}$ are equivalent: The structure axioms for $\mathfrak{C}$ as a †-SMC, as well as the rewriting axioms listed in Figure 1. Note that we identify the basic morphism Id with the identity morphism $1_1 \in \mathfrak{C}(1,1)$ and the basic morphism SWAP as the swap morphism $\sigma_{1,1} \in \mathfrak{C}(2,2)$. Also note that those axioms listed in Figure 1 are only a fragment of all axioms of the ZX-calculus. We omit other axioms for the sake of simplicity. The interested readers can found all axioms of the ZX-calculus in [29,30].

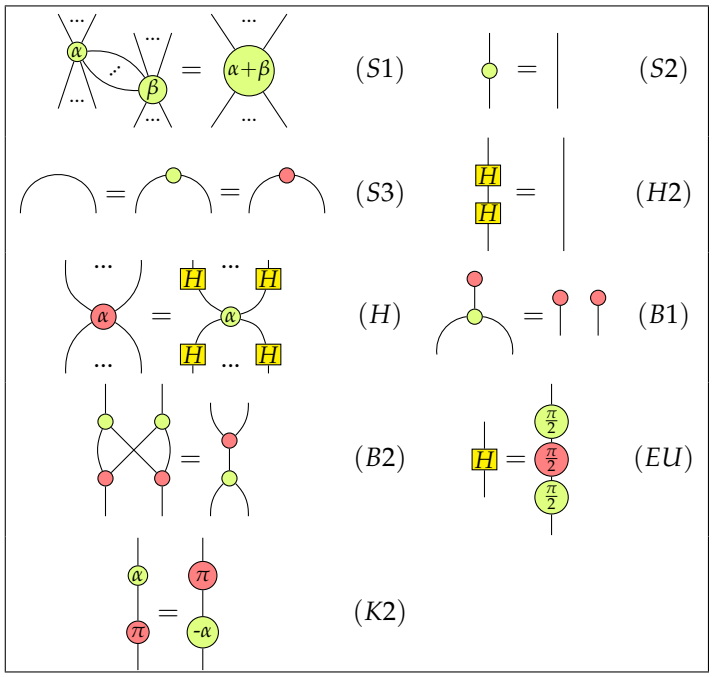


Figure 1. ZX-calculus rules.

2.1.2. Syntax of Clqp

We use $\mathcal{L}_n$ to denote the language of CLQP$_n$. $\mathcal{L}_n$ is defined by the following BNF:

Definition 1 (Language of CLQP$_n$). *For $p \in \mathbb{P}$ and $U \in \mathbb{O}$,*

- $\mathbf{a} := U \mid U^\dagger \mid c_o \mid \phi? \mid \mathbf{a};\mathbf{a} \mid \mathbf{a} \cup \mathbf{a} \mid \mathbf{a}^*$
- $\phi := \top \mid p \mid c_p \mid \neg\phi \mid \phi \wedge \phi \mid [\alpha]\phi \mid K_I\phi \mid P^{\geq r}\phi$

Here c_o and c_p are respectively an operational constant and a propositional constant expressed by the ZX-calculus. More precisely, c_o is a morphism from n to n, while c_p is a morphism from 0 to n. For example, $Z_2^2(\pi)$ is an operational constant of $\mathcal{L}_2$ and $X_0^3(\pi)$ is a propositional constant of $\mathcal{L}_3$. For all $I \subseteq N$ and $I \neq \emptyset$, K_I is an epistemic modality. For all $r \in [0,1]$, $P^{\geq r}$ is a probabilistic modality.

The intended meaning of those formulas is the following:

- $\mathbf{a}$ is a program.
- U is an operational variable that refers to a unitary operation on $\mathbb{C}^{2^n}$.
- $U^\dagger$ is the adjoint of U.
- c_o is an operational constant that refers to a specific operation on $\mathbb{C}^{2^n}$.
- $\phi?$ is the program that refers to the test of proposition ϕ.
- $\mathbf{a}_1; \mathbf{a}_2$ is the sequential composition of $\mathbf{a}_1$ and $\mathbf{a}_2$ (applying first $\mathbf{a}_1$ and then $\mathbf{a}_2$).
- $\mathbf{a}_1 \cup \mathbf{a}_2$ is the non-deterministic choice between of $\mathbf{a}_1$ and $\mathbf{a}_2$ (applying either $\mathbf{a}_1$ or $\mathbf{a}_2$).
- $\mathbf{a}^*$ is the iteration of $\mathbf{a}$, meaning that to repeat $\mathbf{a}$ a finite, but non-deterministically determined, number of times.
- ϕ is a formula.
- $\top$ is a propositional constant representing logical truth.
- p is a propositional variable.
- c_p is a propositional constant that refers to a specific state on $\mathbb{C}^{2^n}$.
- $\neg$ is the classical negation.
- $\wedge$ is the classical conjunction.
- $[\mathbf{a}]\phi$ means that "ϕ will be the case after every execution of $\mathbf{a}$".
- $K_I\phi$ means that "subsystem I carries the information that ϕ is the case".
- $P^{\geq r}\phi$ means that "testing property ϕ (on the current state) will succeed with probability $\geq r$".

We define logical absurdity as $\bot := \neg\top$, classical disjunction as $\phi \vee \psi := \neg(\neg\phi \wedge \neg\psi)$ and quantum negation as $\sim \phi := [\phi?]\bot$. Classical implication and equivalence are respectively defined as $\phi \to \psi := \neg\phi \vee \psi$ and $\phi \leftrightarrow \psi := (\phi \to \psi) \wedge (\psi \to \phi)$. Quantum join is defined as $\phi \sqcup \psi :=\sim (\sim \phi \wedge \sim \psi)$. A range of probabilistic formulas are defined as : $P^{\leq r}\phi := P^{\geq(1-r)} \sim \phi$, $P^{=r}\phi := P^{\geq r}\phi \wedge P^{\leq r}\phi$, $P^{>r}\phi := \neg P^{\leq r}\phi$, $P^{<r}\phi := \neg P^{\geq r}\phi$.

Comparing to LQP, the syntax of CLQP is an extension of LQP with the following additional components: operational constants, propositional constants and the iteration $*$. The iteration $*$ is not included in LQP. We put it back into our logic such that it can be used to verify quantum programs with the while-loop, for example the quantum walk algorithm [31] and the famous HHL (Harrow–Hassidim–Lloyd) quantum algorithms for solving systems of linear equations, which is a cornerstone of quantum machine learning.

2.2. Semantics

The semantics of CLQP$_n$ is based on the following structure.

Definition 2 (Quantum Dynamic Frame [18])**.** *Let $\mathcal{H} = \mathbb{C}^{2^n}$ be the 2^n dimensional Hilbert space. The n-qubit quantum dynamic frame build on $\mathcal{H}$ is the following structure:*

$$\Sigma(\mathcal{H}) := (\Sigma, \{\xrightarrow{S?}\}_{S\in\mathcal{T}}, \{\xrightarrow{U}\}_{U\in\mathcal{U}}).$$

1. *Σ is the set of all one-dimensional subspace of $\mathcal{H}$, called the set of states. We denote a state $s = \overline{x}$ of $\mathcal{H}$ using any of the non-zero vector $x \in \mathcal{H}$ that generates it.*
2. *Call two states s and t orthogonal and write $s \bot t$ if and only if $\forall x \in s$ and $\forall y \in t$, $\langle x|y\rangle = 0$. For a set of states $S \subseteq \Sigma$, we put $S^\bot := \{t \in \Sigma : t \bot s, \forall s \in S\}$ and we denote $\overline{S} = (S^\bot)^\bot$ the biorthogonal closure of S.*
3. *A set of states $S \subseteq \Sigma$ is called a testable property iff it is biorthogonally closed, i.e., $S = \overline{S}$. We denote $\mathcal{T} \subseteq \mathcal{P}(\Sigma)$ the set of all testable properties.*
4. *Every testable property S uniquely corresponds to a subspaces W_S of $\mathcal{H}$ by taking $W_S := \bigcup S$.*

5. *For every testable property S, there is a partial map $S?$ on Σ, called a quantum test, induced by P_{W_S} the projector onto the subspace W_S:*

   - $S?(\overline{x}) := \overline{P_{W_S}(x)} \in \Sigma$, *if* $\overline{x} \notin S^{\perp}$.
   - $S?(\overline{x}) :=$ *undefined, otherwise.*

   We denote by $\xrightarrow{S?} \subseteq \Sigma \times \Sigma$ *the binary relation corresponding to the partial map $S?$, i.e., given by* $s \xrightarrow{S?} t$ *iff* $S?(s) = t$.
6. *$\mathcal{U}$ is the set of all unitary maps on $\mathcal{H}$. For every unitary map U on $\mathcal{H}$, the corresponding binary relation* $\xrightarrow{U} \subseteq \Sigma \times \Sigma$ *is given by* $s \xrightarrow{U} t$ *iff* $U(x) = y$ *for some vector* $x \in s, y \in t$.

An n-qubit quantum dynamic model is a pair $M = (\Sigma(\mathcal{H}), V, R)$, in which $\Sigma(\mathcal{H}) := (\Sigma, \{\xrightarrow{S?}\}_{S \in \mathcal{T}}, \{\xrightarrow{U}\}_{U \in \mathcal{U}})$ is an n-qubit quantum dynamic frame. V is valuation function which maps every $p \in \mathbb{P}$ to a set of states $V(p) \subseteq \Sigma$ and every c_p to a singleton set which contains the special state indicated by c_p. R is an interpretation function that maps quantum programs to relations over Σ. An operational variable U is interpreted by a relation $R(U) = \xrightarrow{U} \subseteq \Sigma \times \Sigma$ induced by a unitary map. An operational constant c_o is interpreted by the relation $R(c_o) = \xrightarrow{c_o} \subseteq \Sigma \times \Sigma$ induced by the unitary map c_o indicates. More precisely, $(s, t) \in R(c_o)$ iff there are $x \in s$ and $y \in t$ such that $\ulcorner c_o \urcorner(x) = y$, where $\ulcorner c_o \urcorner$ is the matrix interpretation of c_o. A test $\phi?$ is interpreted by the relation $R(\phi?) \subseteq \Sigma \times \Sigma$ induced by the quantum projector $P_{W_{\overline{V(\phi)}}}$. The adjoint of a program $U^{\dagger}$ is interpreted by the relation $R(U^{\dagger})$, which is defined by $(s, t) \in R(U^{\dagger})$ iff there are $x \in s$ and $y \in t$ such that $U^{\dagger}(x) = y$. The relational interpretation is extend to arbitrary quantum programs as follows:

- $R(\mathbf{a}_1; \mathbf{a}_2) = R(\mathbf{a}_2) \circ R(\mathbf{a}_1)$.
- $R(\mathbf{a}_1 \cup \mathbf{a}_2) = R(\mathbf{a}_1) \cup R(\mathbf{a}_2)$.
- $R(\mathbf{a}^*) = (R(\mathbf{a}))^*$, i.e., the reflexive transitive closure of $R(\mathbf{a})$.

Definition 3 (Semantics of CLQP$_n$)**.** *Let $M = (\Sigma(\mathcal{H}), V, R)$ be an n-qubit quantum dynamic model. Let $s \in \Sigma$.*

- $M, s \models_n p$ *iff* $s \in V(p)$.
- $M, s \models_n c_p$ *iff* $\{s\} = V(c_p)$.
- $M, s \models_n \neg\phi$ *iff not* $M, s \models_n \phi$.
- $M, s \models_n \phi \wedge \psi$ *iff* $M, s \models_n \phi$ *and* $M, s \models_n \psi$.
- $M, s \models_n [\mathbf{a}]\phi$ *iff for all t, if* $(s, t) \in R(\mathbf{a})$ *then* $M, t \models_n \phi$.
- $M, s \models_n K_I\phi$ *iff for all t, if* $s \sim_I t$ *then* $M, t \models_n \phi$. *Here the relation $\sim_I$ is defined as follows. For all unit vector $x \in s$ and $y \in t$, let $\rho_x = |x\rangle\langle x|$ and $\rho_y = |y\rangle\langle y|$ be the density operator of x and y respectively. Let $tr_{N \setminus I}$ be the partial trace over the environment $N \setminus I$. Then $s \sim_I t$ holds iff $tr_{N \setminus I}(\rho_x) = tr_{N \setminus I}(\rho_y)$.*
- $M, s \models_n P^{\geq r}\phi$ *iff for all unit vector* $x \in s$, $\langle x | P_{W_{\overline{V(\phi)}}} | x \rangle \geq r$.

The semantics of CLQP$_n$ largely coincides with the semantics of LQP: For all formulas that appear in both LQP and CLQP$_n$, their semantics in CLQP are the same as their semantics in LQP. As usual, by $\Phi \models_n \phi$ we mean for all n-qubit quantum dynamic model M and all state s in M, if $M, s \models_n \psi$ for all $\psi \in \Phi$, then $M, s \models_n \phi$. We say that ϕ is valid in CLQP$_n$ if $\emptyset \models_n \phi$.

2.3. Axiomatization

Now we introduce a sound proof system for CLQP. This proof system is an extension of the proof system of LQP with axioms of the ZX-calculus. It consists of the following axioms and rules:

- Axioms of dynamic logic:
  - All propositional tautologies.
  - Kripke Axiom: $\vdash [\mathbf{a}](p \to q) \to ([\mathbf{a}]q \to [\mathbf{a}]q)$.

  - PDL1: $\vdash [\mathbf{a};\mathbf{b}]p \leftrightarrow [\mathbf{b}][\mathbf{a}]p$.
  - PDL2: $\vdash [\mathbf{a} \cup \mathbf{b}]p \leftrightarrow [\mathbf{a}]p \wedge [\mathbf{b}]p$.
  - PDL3: $\vdash [\mathbf{a}^*]p \leftrightarrow (p \wedge [\mathbf{a}][\mathbf{a}^*]p)$.
  - PDL4: $\vdash [\mathbf{a}^*](p \rightarrow [\mathbf{a}]p) \rightarrow (p \rightarrow [\mathbf{a}^*]p)$.
- Axioms of quantum system, in which we use the abbreviations $\langle \mathbf{a} \rangle \phi := \neg[\mathbf{a}]\neg\phi$, $\Box\phi :=\sim \neg\phi$, $\diamond\phi := \neg\Box\neg\phi$.
  - Testability Axiom: $\vdash \Box p \rightarrow [q?]p$.
  - Partial Functionality: $\vdash \neg[p?]q \rightarrow [p?]\neg q$.
  - Adequacy: $\vdash (p \wedge q) \rightarrow \langle p? \rangle q$.
  - Repeatability: $\vdash (\sim\sim p \leftrightarrow p) \rightarrow [p?]p$.
  - Unitary bijectivity 1: $\vdash p \rightarrow [U;U^\dagger]p$.
  - Unitary bijectivity 2: $\vdash p \rightarrow [U^\dagger;U]p$.
  - Proper Superpositions: $\vdash \langle \mathbf{a} \rangle \Box\Box p \rightarrow [\mathbf{b}]p$.
  - Adjointness: $\vdash p \rightarrow [q?]\Box\langle q? \rangle \diamond p$.
- Axioms of the epistemic modality:
  - K: $\vdash K_I(p \rightarrow q) \rightarrow (K_I p \rightarrow K_I q)$.
  - T: $\vdash K_I p \rightarrow p$.
  - S4: $\vdash K_I p \rightarrow K_I K_I p$.
  - S5: $\vdash \neg K_I p \rightarrow K_I \neg K_I p$.
- Axioms of probability:
  - $\vdash P^{\geq 0} p$.
  - $\vdash P^{=1} \top$.
  - $\vdash P^{=0}(p \leftrightarrow\sim p)$.
  - $\vdash (p \leftrightarrow q) \rightarrow (P^{=r} p \leftrightarrow P^{=r} q)$.
  - $\vdash \Box\Box(p \rightarrow\sim q) \rightarrow (P^{=r}(p \sqcup q) \rightarrow (P^{=s} p \rightarrow P^{r-s} q))$.
  - $\vdash (\Box\Box(p \rightarrow q) \wedge P^{=r} q \wedge [q?]P^{=s} p) \rightarrow P^{=rs} p$.
  - $\vdash (\Box\Box(p \rightarrow q) \wedge P^{>0} p \wedge P^{>0} q) \rightarrow P^{>0}(P^{=r} p \wedge P^{=1-r} q)$.
- Rules:
  - Modus Ponens (MP): $\frac{\phi \quad \phi \rightarrow \psi}{\psi}$.
  - Necessitation (Nec): $\frac{\phi}{[\mathbf{a}]\phi}$.
  - Uniform Substitution (US): $\frac{\phi(p)}{\phi(q/p)}$.
  - ZX equivalence: if $\mathbf{a}_1 = \mathbf{a}_2$ in the ZX-calculus, then $\vdash [\mathbf{a}_1]p \leftrightarrow [\mathbf{a}_2]p$.

Theorem 1. *All axioms and rules above are sound with respect to the semantics of $CLQP_n$, for all $n \geq 1$.*

Proof. (sketch) Axioms of dynamic logic are valid because they are valid in every dynamic logic and CLQP is a special dynamic logic. The validity of axioms of quantum systems is shown in [17,18]. Axioms of epistemic logic are valid because $\sim_I$ is an equivalence relation. The validity of axioms of probability can be found in [21]. The rules MP and US are valid in all logical systems. The rule Nec is valid because $[\mathbf{a}]$ is a necessity modality. The rule ZX equivalence is valid because if $\mathbf{a}_1 = \mathbf{a}_2$ in the ZX-calculus, then by the soundness of the ZX calculus we know they represent the same linear map. □

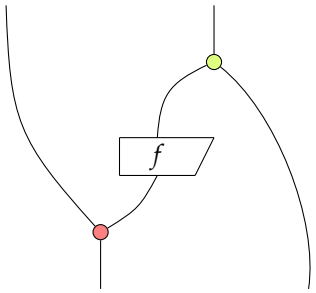
Remark 1. *In $CLQP_1$, axioms of the ZX-calculus are simplified. Those axioms of single-qubit ZX-calculus can be found in Backens [32].*

3. CLQP for Verification

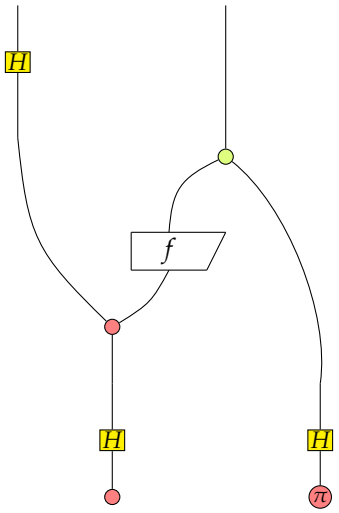
In this section we are going to demonstrate the usage of CLQP by applying it to the formal verification of quantum programs and protocols. For the sake of simplicity, we choose to verify the correctness of Deutsch's algorithm and the concealing property of quantum bit commitment protocols.

3.1. Deutsch'S Algorithm

Deutsch's algorithm is a simple algorithm that solves a slightly contrived problem [33]. It determines whether a function f from $\{0,1\}$ to $\{0,1\}$ is constant or balanced, where f being constant means that $f(0) = f(1)$ and balanced otherwise. We can formalize Deutsch's algorithm in CLQP_2. First, as it is shown in Chapter 12 of [26], we build an oracle U_f as the following:

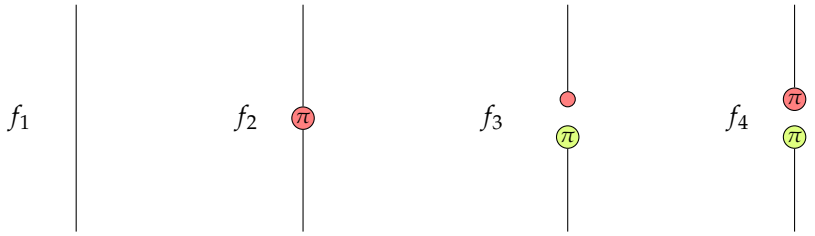


Formally, we have $U_f = (I \otimes Z_2^1(0)) \circ (I \otimes f \otimes I) \circ (X_1^2(0) \otimes I)$. It holds that for all $f : \{|0\rangle, |1\rangle\} \to \{|0\rangle, |1\rangle\}$, we have $U_f(|x, y\rangle) = |x\rangle|y \oplus f(x)\rangle$. The unitary operation of Deutsch's Algorithm is graphically represented as the following:



After this operation, a measurement on the $\{|0\rangle, |1\rangle\}$ basis is applied to the first qubit. The function f is constant if and only if the result of the measurement is $|0\rangle$.

There are four functions from $\{|0\rangle, |1\rangle\}$ to $\{|0\rangle, |1\rangle\}$. Let them be f_1, f_2, f_3, f_4, where $f_1(|0\rangle) = |0\rangle, f_1(|1\rangle) = |1\rangle, f_2(|0\rangle) = |1\rangle, f_2(|1\rangle) = |0\rangle, f_3(|0\rangle) = f_3(|1\rangle) = |0\rangle, f_4(|0\rangle) = f_4(|1\rangle) = |1\rangle$. Note that up to a non-zero scalar, we have $f_1 = X_1^1(0), f_2 = X_1^1(\pi), f_3 = X_0^1(0) \circ Z_1^0(\pi), f_4 = X_0^1(\pi) \circ Z_1^0(\pi)$.



The correctness of Deutsch's Algorithm can be expressed by the following deduction:

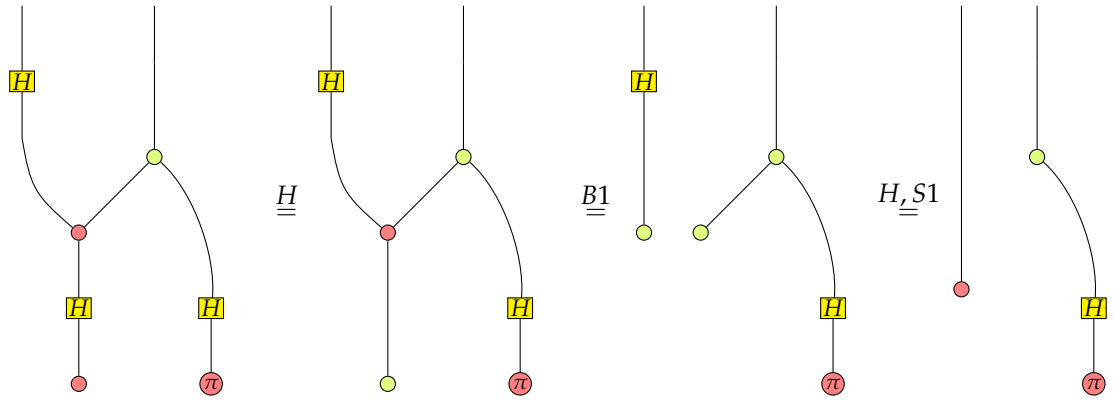
- $X_0^1(0) \otimes X_0^1(\pi) \vDash_2 [(H \otimes I) \circ U_f \circ (H \otimes H)]P^{=1}K_1(X_0^1(0))$ for all constant function f.
- $X_0^1(0) \otimes X_0^1(\pi) \vDash_2 [(H \otimes I) \circ U_f \circ (H \otimes H)]P^{=0}K_1(X_0^1(0))$ for all balanced function f.

Equivalently, it can be characterized by the validity of the following formula of CLQP_2:

$$(X_0^1(0) \otimes X_0^1(\pi)) \rightarrow ([(H \otimes I) \circ (U_{f_1} \cup U_{f_2}) \circ (H \otimes H)]P^{=1}K_1(X_0^1(0)))\wedge,$$
$$(X_0^1(0) \otimes X_0^1(\pi)) \rightarrow ([(H \otimes I) \circ (U_{f_3} \cup U_{f_4}) \circ (H \otimes H)]P^{=0}K_1(X_0^1(0))).$$

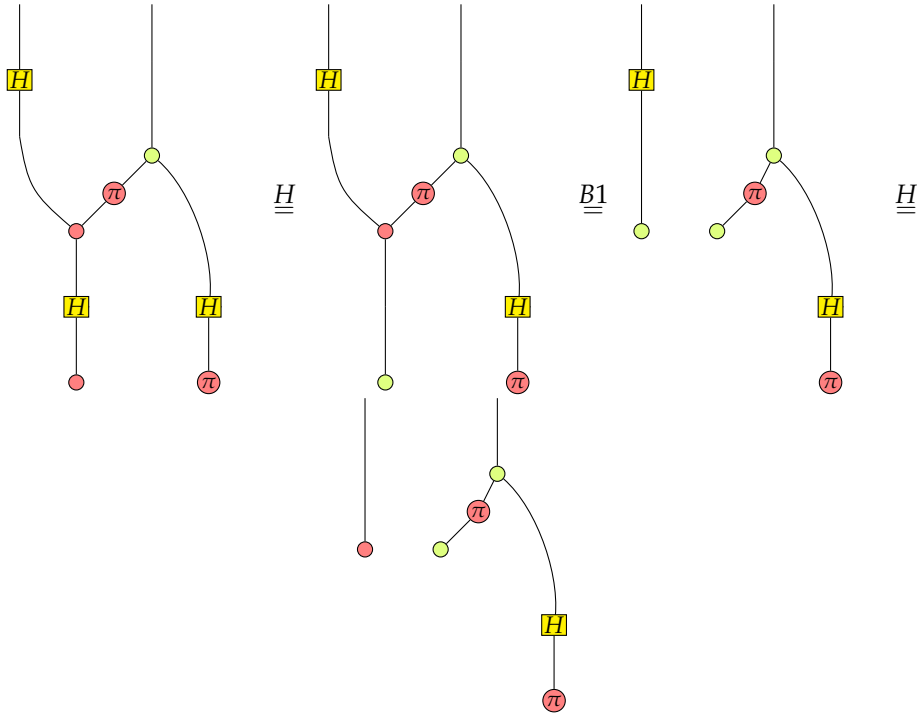
The graphical representation of the ZX-calculus can significantly simplify the proof of the validity of the above formula.

- Suppose the function is f_1, then we have



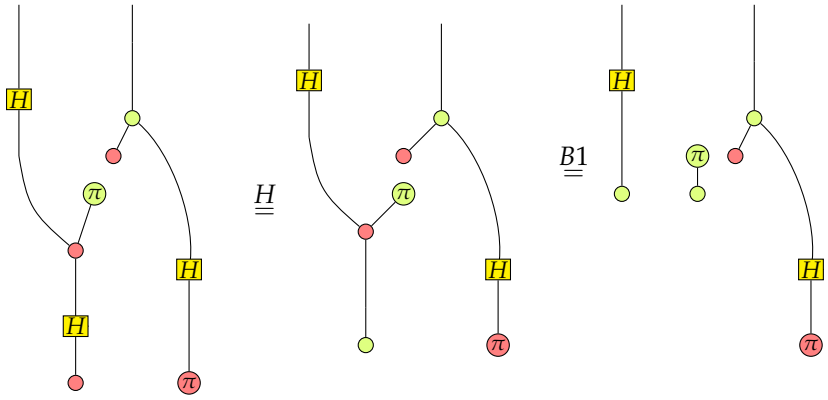
From the rightmost graph we know that $P^{=1}K_1(X_0^1(0))$ is satisfied the first qubit is in the $X_0^1(0)$ state.

- Suppose the function is f_2, then we have



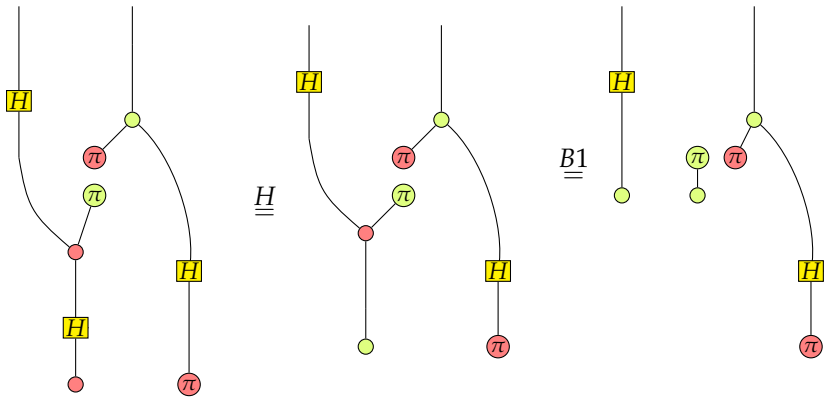
Apparently, $P^{=1}K_1(X_0^1(0))$ is satisfied in the rightmost graph.

- Suppose the function is f_3, then we have



Note that $Z_1^0(\pi) \circ Z_1^0(0)$ is 0. We know that $P^{=0}K_1(X_0^1(0))$ is satisfied in the rightmost graph.

- Suppose the function is f_4, then we have



Since the absorbing scalar 0 is in the rightmost graph, we know that $P^{=0}K_1(X_0^1(0))$ is satisfied.

3.2. Quantum Bit Commitment

A bit commitment protocol consists of two phases: commit and opening. In the commit phase, Alice the sender chooses a bit a ($a = 0$ or 1) which she wishes to commit to the receiver Bob. Then Alice presents Bob some evidence about the bit. The committed bit cannot be known by Bob prior to the opening phase. Later, in the opening phase, Alice announces some information for reconstructing a. Bob then reconstructs a bit a' using Alice's evidence and announcement. A correct bit commitment protocol will ensure that $a' = a$. A bit commitment protocol is concealing if Bob cannot know the bit Alice committed before the opening phase and it is binding if Alice cannot change the bit she committed after the commit phase.

Quantum bit commitment (QBC) was first proposed by Bennett and Brassard in 1984 [34]. In a QBC protocol, quantum operation and communication are used to ensure the concealing and binding property. In the literature [35–37], it is acknowledged that a general model of QBC protocols should at least includes the following ingredients:

1. The Hilbert space required to describe the protocol is the tensor product of the Hilbert spaces that play a role in the protocol.
2. The total system is initially in a pure state.
3. Every action taken by a party corresponds to that party performing a unitary operation on the systems in his/her possession.
4. Every communication corresponds to a party sending a subset of the systems in his/her possession to the other party.

Bearing this common knowledge in mind, a rigorous and simple formalization of quantum bit commitment is given as follows.

Definition 4. *A quantum bit commitment protocol consists of the following:*

1. *Two finite dimensional Hilbert spaces A and B.*
2. *Two pure states $|L\rangle, |R\rangle \in A \otimes B$.*
3. *A completely positive map $Open$ on $A \otimes B$ such that $Open(|L\rangle\langle L|)$ is orthogonal to $Open(|R\rangle\langle R|)$.*

This formalization provides a high level description of quantum bit commitment. Initially, Alice (possibly with the help of Bob) prepares a state $|L\rangle$ or $|R\rangle$ of quantum system $A \otimes B$ depending on the value of Alice's bit. (Note that $|L\rangle$ and $|R\rangle$ are not the initial state of the QBC protocol, but the final state of the commit phase. Starting from a pure state, a commit phase may involve many rounds of actions and communications). Alice sends $Tr_A(|L\rangle\langle L|)$ or $Tr_A(|R\rangle\langle R|)$ to Bob to perform the commitment. At the opening stage, Alice sends the rest sub-state of $|L\rangle$ or $|R\rangle$ to Bob to allow him to verify her commitment. Bob applies the completely positive map $Open$ to determine Alice's commitment. The QBC protocol is concealing if $Tr_A(|L\rangle\langle L|) = Tr_A(|R\rangle\langle R|)$. It is binding if there is no unitary map U on A such that $(U \otimes I)|L\rangle = |R\rangle$.

In CLQP, the concealing property of the QBC protocol can be characterized by the validity of the following formula:

$$K_B c_L \leftrightarrow K_B c_R.$$

Here c_L and c_R are respectively the propositional constant that characterized the state $|L\rangle$ and $|R\rangle$. The universality of the ZX calculus ensures that c_L and c_R can be characterized in CLQP. The validity of this formula implies that $\overline{\{|L\rangle\}} \sim_B \overline{\{|R\rangle\}}$, which further entails that $Tr_A(|L\rangle\langle L|) = Tr_A(|R\rangle\langle R|)$. It seems the binding property cannot be characterized by formulas of CLQP. However, we can still use the ZX-calculus to prove non-binding since the ZX-calculus is universal and sound.

4. Conclusions and Future Work

In this paper we introduce the basic ideas of the categorical logic of quantum programs. We present the syntax, semantics and proof system of this logic and demonstrate its usage in the formal verification of quantum programs and protocols. In a nutshell, CLQP is an extension of LQP with a universal set of constant symbols and iteration $*$.

Our long-term goal is to develop CLQP as a powerful tool for the verification of quantum programs and protocols. In the recent future, we will study the decidability and complexity of CLQP. We will also apply CLQP to the formal verification of more complicated quantum programs and protocols, for example the HHL algorithm in quantum machine learning. The semantics of CLQP presented in this paper is based on pure quantum states. The development of mixed-state semantics is in our agenda.

Author Contributions: Conceptualization, X.S. and F.H.; Methodology, X.S.; Validation, F.H.; Formal analysis, X.S. and F.H.; Writing—original draft preparation, X.S.; Writing—review and editing, X.S. and F.H.; Funding acquisition, X.S. All authors have read and agreed to the published version of the manuscript.

Funding: The project is funded by the Minister of Science and Higher Education within the program under the name"Regional Initiative of Excellenc" in 2019-2022, project number: 028/RID/2018/19, the amount of funding: 11 742 500 PLN.

Acknowledgments: The authors are grateful to Xishun Zhao for daily discussion.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A. Dagger Symmetric Monoidal Category

Definition A1 (Category). *A category $\mathcal{C}$ consists of:*

1. *a collection $ob(\mathcal{C})$ of objects,*
2. *for every pair of objects A, B, a set $\mathcal{C}(A, B)$ of morphisms,*
3. *for every object A, a special identity morphism: $1_A \in \mathcal{C}(A, A)$,*
4. *sequential composition operation for morphisms:*

$$\circ : \mathcal{C}(B, C) \times \mathcal{C}(A, B) \to \mathcal{C}(A, C),$$

satisfying the following conditions:

1. *$\circ$ is associative on morphisms: $(h \circ g) \circ f = h \circ (g \circ f)$,*
2. *$\circ$ is unital on morphisms: $1_B \circ f = f = f \circ 1_A$, for all $f \in \mathcal{C}(A, B)$.*

Definition A2 (Strict Monoidal Category [26]). *A strict monoidal category $\mathcal{C}$ is a category equipped with:*

1. *a parallel composition operation for objects:*

$$\otimes : ob(\mathcal{C}) \times ob(\mathcal{C}) \to ob(\mathcal{C}),$$

2. *a unit object $I \in ob(\mathcal{C})$,*
3. *and a parallel composition operation for morphisms:*

$$\otimes : \mathcal{C}(A, B) \times \mathcal{C}(C, D) \to \mathcal{C}(A \otimes C, B \otimes D)$$

satisfying the following conditions:

1. *$\otimes$ is associative and unital on objects:*

$$(A \otimes B) \otimes C = A \otimes (B \otimes C) \qquad A \otimes I = A = I \otimes A,$$

2. *$\otimes$ is associative and unital on morphisms:*

$$(f \otimes g) \otimes h = f \otimes (g \otimes h) \qquad f \otimes 1_I = f = 1_I \otimes f,$$

3. *$\otimes$ and $\circ$ satisfy the interchange law:*

$$(g_1 \otimes g_2) \circ (f_1 \otimes f_2) = (g_1 \circ f_1) \otimes (g_2 \circ f_2).$$

Definition A3 (Symmetric Monoidal Category [26]). *A strict monoidal category is symmetric if it is equipped with a swap morphism:*

$$\sigma_{A,B} : A \otimes B \to B \otimes A$$

defined for all objects A, B, satisfying:

- $\sigma_{B,A} \circ \sigma_{A,B} = 1_{A \otimes B}$,
- $\sigma_{A,I} = 1_A$,
- $(g \otimes f) \circ \sigma_{A,B} = \sigma_{A',B'} \circ (f \otimes g)$, *for all* $f \in \mathcal{C}(A, A')$, $g \in \mathcal{C}(B, B')$,
- $(1_B \otimes \sigma_{A,C}) \circ (\sigma_{A,B} \otimes 1_C) = \sigma_{A,B \otimes C}$.

Definition A4 (Dagger Functor †[26]). *A dagger functor for a strict monoidal category is an operation † that satisfy the following:*

- *identity on objects: $A^\dagger = A$,*
- *reserves morphisms: $(f : A \to B)^\dagger := f^\dagger : B \to A$,*
- *is involutive: $(f^\dagger)^\dagger = f$,*
- *and respects the symmetric monoidal category structure:*

$$(g \circ f)^\dagger = f^\dagger \circ g^\dagger \qquad (f \otimes g)^\dagger = f^\dagger \otimes g^\dagger.$$

A dagger symmetric monoidal category is a strict monoidal category equiped with a dagger functor.

Reference

1. Feng, Y.; Duan, R.; Ji, Z.; Ying, M. Probabilistic bisimulations for quantum processes. *Inf. Comput.* **2007**, *205*, 1608–1639. [CrossRef]
2. Ying, M.; Feng, Y.; Duan, R.; Ji, Z. An algebra of quantum processes. *ACM Trans. Comput. Log.* **2009**, *10*, 19. [CrossRef]
3. Feng, Y.; Duan, R.; Ying, M. Bisimulation for Quantum Processes. *ACM Trans. Program. Lang. Syst.* **2012**, *34*, 17. [CrossRef]
4. Kubota, T.; Kakutani, Y.; Kato, G.; Kawano, Y.; Sakurada, H. Semi-automated verification of security proofs of quantum cryptographic protocols. *J. Symb. Comput.* **2016**, *73*, 192–220. [CrossRef]
5. Birkhoff, G.; von Neumann, J. The Logic of Quantum Mechanics. *Ann. Math.* **1936**, *37*, 823–843. [CrossRef]
6. Piron, C. Axiomatique quantique. *Helv. Phys. Acta* **1964**, *37*, 439–468.
7. Malinowski, J. Quantum experiments and the lattice of orthomodular logics. *Log. Anal.* **1999**, *35*, 165–166.
8. Malinowski, J. On the lattice of orthomodular logics. *Bull. Sect. Log.* **1999**, *28*, 11–18.
9. Ledda, A.; Konig, M.; Paoli, F.; Giuntini, R. MV-Algebras and Quantum Computation. *Stud. Log.* **2006**, *82*, 245–270. [CrossRef]
10. Giuntini, R.; Ledda, A.; Paoli, F. Expanding Quasi-MV Algebras by a Quantum Operator. *Stud. Log.* **2007**, *87*, 99–128. [CrossRef]
11. Gudder, S. Quantum Computational Logic. *Int. J. Theor. Phys.* **2003**, *42*, 39–47. [CrossRef]
12. Dunn, J.M.; Moss, L.S.; Wang, Z. Editors' Introduction: The Third Life of Quantum Logic: Quantum Logic Inspired by Quantum Computing. *J. Philos. Log.* **2013**, *42*, 443–459. [CrossRef]
13. Brunet, O.; Jorrand, P. Dynamic quantum logic for quantum programs. *Int. J. Quantum Inf.* **2004**, *2*, 45–54. [CrossRef]
14. Chadha, R.; Mateus, P.; Sernadas, A. Reasoning About Imperative Quantum Programs. *Electron. Notes Theor. Comput. Sci.* **2006**, *158*, 19–39. [CrossRef]
15. Feng, Y.; Duan, R.; Ji, Z.; Ying, M. Proof rules for the correctness of quantum programs. *Theor. Comput. Sci.* **2007**, *386*, 151–166. [CrossRef]
16. Ying, M. Floyd-hoare logic for quantum programs. *ACM Trans. Program. Lang. Syst.* **2011**, *33*, 19:1–19:49. [CrossRef]
17. Baltag, A.; Smets, S. Complete Axiomatizations for Quantum Actions. *Int. J.Theor. Phys.* **2005**, *44*, 2267–2282. [CrossRef]
18. Baltag, A.; Smets, S. LQP: The dynamic logic of quantum information. *Math. Struct. Comput. Sci.* **2006**, *16*, 491–525. [CrossRef]
19. Baltag, A.; Smets, S. A Dynamic-Logical Perspective on Quantum Behavior. *Stud. Log.* **2008**, *89*, 187–211. [CrossRef]
20. Baltag, A.; Bergfeld, J.; Kishida, K.; Sack, J.; Smets, S.; Zhong, S. PLQP & Company: Decidable Logics for Quantum Algorithms. *Int. J. Theor. Phys.* **2014**, *53*, 3628–3647. [CrossRef]
21. Bergfeld, J.M.; Sack, J. Deriving the correctness of quantum protocols in the probabilistic logic for quantum programs. *Soft. Comput.* **2017**, *21*, 1421–1441. [CrossRef]
22. Rad, S.R.; Shirinkalam, E.; Smets, S. A Logical Analysis of Quantum Voting Protocols. *Int. J. Theor. Phys.* **2017**, *56*, 3991–4003. [CrossRef]
23. Abramsky, S.; Coecke, B. A Categorical Semantics of Quantum Protocols. In Proceedings of the 19th IEEE Symposium on Logic in Computer Science (LICS 2004), Turku, Finland, 14–17 July 2004; pp. 415–425. [CrossRef]
24. Coecke, B.; Duncan, R. Interacting quantum observables: Categorical algebra and diagrammatics. *New J. Phys.* **2011**, *13*, 1–85. [CrossRef]
25. Coecke, B.; Heunen, C.; Kissinger, A. Categories of quantum and classical channels. *Quantum Inf. Process.* **2016**, *15*, 5179–5209. [CrossRef]
26. Coecke, B.; Kissinger, A. *Picturing Quantum Processes: A First Course in Quantum Theory and Diagrammatic Reasoning*; Cambridge University Press: Cambridge, UK, 2017.
27. Coecke, B.; Duncan, R. Interacting Quantum Observables. In *International Colloquium on Automata, Languages, and Programming*; Springer: Berlin/Heidelberg, Germany, 2008; Volume 5126, pp. 298–310. [CrossRef]

28. Backens, M. The ZX-calculus is complete for stabilizer quantum mechanics. *New J. Phys.* **2014**, *16*, 093021. [CrossRef]
29. Jeandel, E.; Perdrix, S.; Vilmart, R. A Complete Axiomatisation of the ZX-Calculus for Clifford+T Quantum Mechanics. In Proceedings of the Thirty-Third Annual ACM/IEEE Symposium on Logic in Computer Science, Oxford, UK, 12 July 2018.
30. Hadzihasanovic, A.; Ng, K.F.; Wang, Q. Two complete axiomatisations of pure-state qubit quantum computing. In Proceedings of the Thirty-Third Annual ACM/IEEE Symposium on Logic in Computer Science, Oxford, UK, 12 July 2018.
31. Ambainis, A. Quantum walks and their algorithmic applications. *Int. J. Quantum Inf.* **2003**, *1*, 507–518. [CrossRef]
32. Backens, M. The ZX-calculus is complete for the single-qubit Clifford+T group. In Proceedings of the 12th International Conference on Quantum Physics and Logic, Oxford, UK, 15–17 July 2015.
33. Yanofsky, N.; Mannucci, M. *Quantum Computing for Computer Scientists*; Cambridge University Press: Cambridge, UK, 2008.
34. Bennetta, C.; Brassard, G. Quantum cryptography: Public key distribution and coin tossing. In Proceedings of the IEEE International Conference on Computers, Systems and Signal Processing, Bangalore, India, 9–12 December 1984; pp. 175–179.
35. Mayers, D. Unconditionally secure quantum bit commitment is impossible. *Phys. Rev. Lett.* **1997**, *78*, 3414–3417. [CrossRef]
36. Lo, H.K.; Chau, H.F. Is Quantum Bit Commitment Really Possible? *Phys. Rev. Lett.* **1997**, *78*, 3410–3413. [CrossRef]
37. Spekkens, R.W.; Rudolph, T. Degrees of concealment and bindingness in quantum bit commitment protocols. *Phys. Rev. A* **2001**, *65*, 012310. [CrossRef]